

Efficient Representation of Smart Environments using a Parallelized Approach

Rajalakshmi V, Susan Elias

Sri Venkateswara College of Engineering/Department of Computer Science and Engineering, Chennai, India
vraji@svce.ac.in, susan@svce.ac.in

Abstract—A parallel approach to model Smart Environments in order to represent them efficiently has been presented in this paper. The focus of this research work is to achieve fast response during the dynamic retrieval of information in smart environments. The motivation here is to enable assisted living in an effective manner in smart home environments. Representation of smart environments includes the representation of the real-time data as well as representation of the system. Real-time data from the smart environment is represented here using the separate chaining hash table data structure while the smart environment is represented here using a variant of an existing membrane computing model referred to as the Distributed *P* System. The model proposed in this research work is designed to produce alerts spontaneously in the environment when abnormal activities are identified by a reasoning system that is assumed to be part of the smart environment. A prototype of the above model has been developed and it is capable of generating SMS alerts that can be sent to the occupant's care taker for remote monitoring of the occupant.

Index Terms—Smart Environment, Membrane Computing, Distributed *P* Systems, Assisted Living, Hashing, Parallelization

I. OVERVIEW

Representing smart environments [1][2] is as of today one of the least explored areas, with new options opening up as research challenges. Effective representation will pave the way for effective reasoning, which indicates that the environment is intelligent [3][4] enough to think/learn and even assist occupants in several ways. No such definite model has been proposed yet. This research work aims to create one such state of the art model, which satisfies the aforementioned criteria. Representation of smart environment involves representing system and data. The smart environment is considered to be a system which is represented here using a variant of the existing membrane computing model, the Distributed *P* System [5], which is a collection of heterogeneous *P* systems. The system considered here is an assisted living scenario for elders, which monitors the occupants continuously and alerts them in case of abnormal events by giving them alarms. Evolution rules-rules that govern the mechanism and functionality of the system are required for effective modeling. Data from the smart environment is represented using hash table data structure. Data includes all the occupant activities carried out in the environment. The search routine is designed which compares the live data with the existing rules, if there is a

match, and then an alert is generated. Algorithms that specify the steps involved in the automatic creation of *P* systems as well as the Distributed *P* system to model the smart environment, had been proposed in [6] and presented here for clarity in the proposed concept. This research work also presents experimental results obtained by a parallel implementation of the search routine that enables efficient and fast decision making within the smart environment. In Section 2 the proposed work that focuses on the representation of real-time data is presented and in Section 3 its incorporation in the model of the smart environment is dealt with. The experimental results are also presented before the conclusion of the paper in Section 4.

II. PROPOSED WORK

Prototype is designed to collect the occupant activity recorded by the smart devices. The live data collected is converted to an intermediate form which contains the details of the smart device, the activity and the level of the activity, and also the temporal and spatial data of the activity. These data are stored on the hash table data structure. Distributed *P* system that has already been populated with dynamic rules will perform computations on the data stored on the data structure. They perform comparison with the existing rules. If the rule exists, then the alert/warning is triggered in order to alarm the occupant. The alert is sent as SMS to the care taker of the occupant. Proposed Work is illustrated in "Fig. 1".

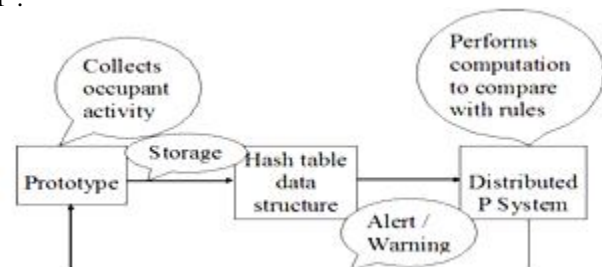


Fig. 1. Proposed System

A. Representaion of System

System here refers to the smart environment. The set of all abnormal activities of the occupant in the system are stored as rules in the Distributed *P* system. The user interface of the prototype collects all the abnormal activities of the occupant from the user. These inputs are converted as rules and stored onto the Distributed *P* system which is illustrated in "Fig. 2".

B. Representaion of Data

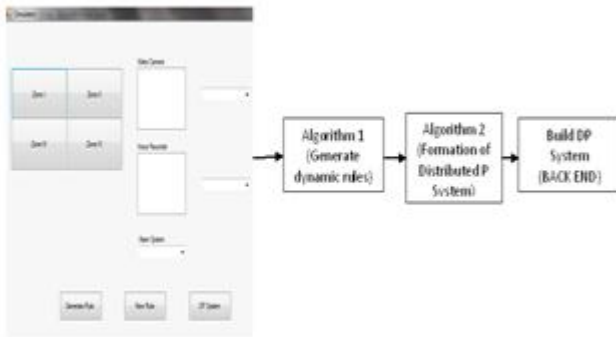


Fig. 2. Representation of system

As the live data is collected from the smart device, the user interface of the prototype represents this data in the form of Spatial-Temporal (*ST*) relation. Spatio-Temporal relation is the relation that is generated for every spatial change in the environment. Spatial change is caused by the occupant movement. For each spatial change, a set of *ST* relations are generated with respect to each object in that particular zone. These data are stored in the Separate Chaining hash table. Representation of data is illustrated in “Fig. 3”.

The Spatio-Temporal relation is of the form as stated below.

$ST(stid, A, dT, Time, X, Y, dist)$

where,

- stid* - identifier for the *ST* relation
- A* - name of the object
- dT* - difference in time in seconds
- Time* - time of movement
- X* - *x* coordinate of the object
- Y* - *y* coordinate of the object
- dist* - distance between *A* and Occupant

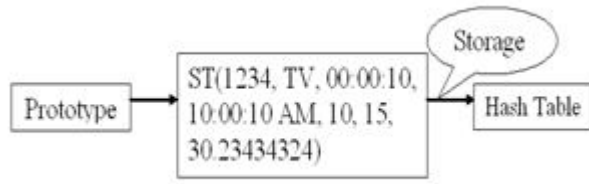


Fig. 3. Representation of data

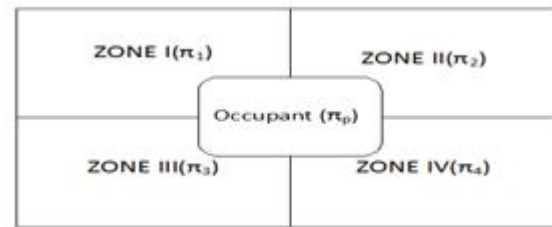
III. PROPOSED SYSTEM MODEL

A. Variant of Distributed *P* System

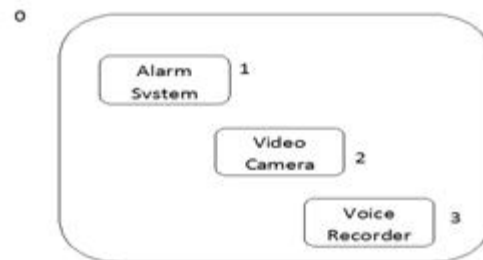
Distributed *P* system is a collection of heterogeneous *P* systems. The formalism of existing *dP* system is modified slightly to develop a variant of it in order to suit the needs of the proposed system. The work proposed in this project work utilizes a variant of the Distributed *P* system for dynamic modeling of smart environment to enable their effective representation. The following subsection presents the formal definition of Distributed *P* systems and the algorithms proposed for their dynamic generation. In order to take a domain specific application, we consider a smart home that can be designed specifically for assisted living. There are several instances where senior citizens live alone in their homes. The design of a smart home that can monitor their activities and guide them suitably would be a great

contribution to an ever increasing social need.

Consider the environment to be a home and assume that there is a single occupant residing in it. This area needs to be embedded with a monitoring system that can reason out and respond suitably, according to a particular situation. The entire environment is represented as a Distributed *P* system. The environment is divided into zones where each zone is a *P* system and a collection of such *P* systems form a Distributed *P* system. This is illustrated in “Fig. 4”, where four zones are identified in the environment which is represented as four *P* systems Π_1, Π_2, Π_3 and Π_4 respectively. The single occupant in the home is considered as a mobile *P* system uniquely labeled as Π_p .

Fig. 4. Smart Home as Distributed *P* System

The devices that enable monitoring could range from video cameras and audio recorders to mikes, sensors (light and smoke), controllers, LEDs etc., The devices used for this application are assumed to be smart. Smart devices are the devices that not only have the ability to capture data but also apply reasoning on the data being captured in order to sense the occurrence of an event. Each of these devices will form the membrane of the *P* system that represents a zone where the devices are physically located. Evolution rules are written to trigger the alarm in each zone in case of an emergency, and an algorithm was proposed to generate them automatically, instead of manually configuring them.

Fig. 5. The *P* System representing a zone

A prototype is developed, that helps in the design of smart spaces. The reason for employing the Distributed *P* system at the back end is to enable model checking, verification and testing of the design of the smart environment prior to its implementation. “Fig. 5”, is an example of *P* system with membranes indicating the presence of smart devices in a particular zone.

The formal definition of proposed variant of Distributed *P* System with degree ($n \geq 1$) has the following construct:

$$\Delta = (O, \mathcal{D}_1, \dots, \mathcal{D}_n, \mathcal{D}_p, S);$$

where,

1. *O* is an alphabet of objects;
2. $\mathcal{D}_1, \dots, \mathcal{D}_n$ are *P* Systems containing *m* membranes with skin

membranes labeled with s_p, \dots, s_n , respectively where,

$$\mathcal{D}_i = (V_p, \mu_p, w_{ip}, \dots, w_{im}, E_p, R_{ip}, \dots, R_{im})$$

where,

- V_i is an alphabet of objects, $V_i \subseteq O$;
 - μ_i is a membrane structure of the i^{th} P System which is of the form $[_0[_1[_2[_3 \dots [_m]_m]_0]$;
 - w_{ip}, \dots, w_{im} represents the multisets of objects available in each membrane;
 - $E_i \subseteq V_i$ represents the objects available in arbitrarily many copies in the environment;
 - R_{ip}, \dots, R_{im} represents the evolution and communication rules used in each P System. The rules have the form $a \rightarrow v$, where $a, v \in V_i$.
3. \mathcal{D}_p is a mobile P System that represents the single occupant in the environment. This is an additional component proposed in this paper to suit the domain specific application. \mathcal{D}_p has the following construct:

$$\mathcal{D}_p = (V_p, \mu_p, w_p, E_p, R_p)$$

where,

- V_p is an alphabet of objects, $V_p \subseteq O$;
 - μ_p is a membrane structure of the mobile P System with only the skin membrane $[_0]_0$;
 - w_p are strings representing the multisets over V_p associated with skin region;
 - $E_p \subseteq V_p$ represents the objects available in arbitrarily many copies in the environment;
 - R_p represents evolution rules of the mobile P System.
4. S is a finite set of rules of the form $(s_p, u/v, s_j)$,

$$1 \leq i, j \leq n, i \neq j \text{ and } u, v \in O^*$$

B. Algorithms

Algorithms for formation of P System and Distributed P System presented in [6] are included here to explain the extension of the work in representing real-time data. The following *Algorithm* will generate rules for triggering the alarm. This algorithm collects input from the prototype. Output of this algorithm is a rule generated to trigger an alarm system.

Input: $i, \text{Dev}_1, \text{Dev}_2, \dots, \text{Dev}_m$

Output: A rule generated to trigger the alarm system

- Set LHS = RHS = NULL
- for each membrane $k = 1$ in \mathcal{D}_i do
- $V_i \leftarrow V_i \cup \{lab_{ik}\}$
- $w_{ik} \leftarrow w_{ik} \cup \{lab_{ik}^{lev}\}$
- $R_{ik} \leftarrow R_{ik} \cup \{lab_{ik}^{lev} \rightarrow (lab_{ik}^{lev}, out)\}$
- $LHS \leftarrow LHS \cup \{lab_{ik}^{lev}\}$
- end
- For membrane $k = I$,
- $V_i \leftarrow V_i \cup \{lab_{iI}\}$
- $w_{iI} \leftarrow w_{iI} \cup \{lab_{iI}^{lev}\}$
- $RHS \leftarrow lab_{iI}^{lev}$
- $R_{iI} \leftarrow R_{iI} \cup \{LHS \rightarrow RHS\}$
- $R_{iI} \leftarrow R_{iI} \cup \{RHS \rightarrow s_i\}$
- $R_{iI} \leftarrow R_{iI} \cup \{s_i \rightarrow lab_{jI}^{lev}\}$
- $w_i \leftarrow w_{ik}$

The following *Algorithm* generates the Distributed P System. This algorithm takes as input all individual P Systems to form Distributed P System. Assume i, j and k represents the P System.

Input: P Systems = $(\Pi_1, \dots, \Pi_n, \Pi_p)$

Output: DP System $\Delta = (O, \Pi_1, \dots, \Pi_n, \Pi_p, S)$

- Set $O \leftarrow \emptyset$
- for each Π_i do
- $O \leftarrow O \cup \{V_i\}$
- $\Delta \leftarrow \Delta \cup \{\Pi_i\}$
- $S \leftarrow S \cup \{(s_p, u/\lambda, s_j)\}$
- end
- For P System Π_p ,
- $\Delta \leftarrow \Delta \cup \{\Pi_p\}$
- $S \leftarrow S \cup \{(s_p, u/\lambda, s_p)\}$

C. Separate Chaining Hash Table

The Separate Chaining hash table contains all the occupant movement. The hash table is indexed with respect to the coordinates of the space. Consider the environment space is 400×400 . The hash table contains entries from coordinates (1,1) to (400,400). Hash table structure is illustrated in the "Fig. 6".

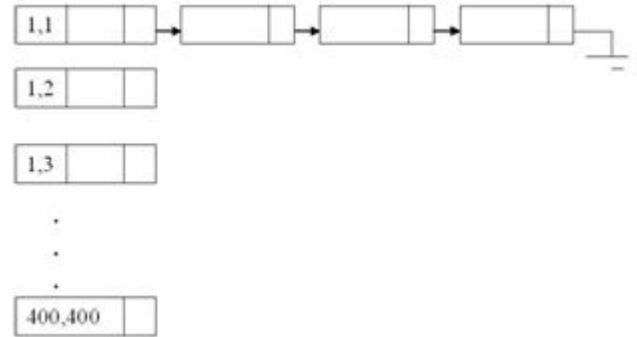


Fig. 6. Separate Chaining Hash table Structure

Each entry in the hash table is a pointer to a linked list. Each node in the linked list contains the occupant movement hashed to the same coordinates which is shown in "Fig. 7".

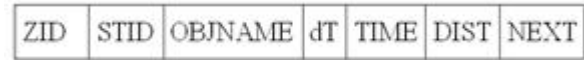


Fig. 7. Linked list node Structure

After storing the data in the hash table, the search routine is invoked to compare the activity with the rules in the Distributed P System. If the activity is an anomaly, then the alert is generated. The alert is also sent as SMS.

D. Parallelization of Search Module

Search module compares the hash table entry with the existing rules of the Distributed P System. Hash table contains all occupant movement. In this module, comparison is done by single processor. By parallelizing the search module, the response time will be improved. While parallelizing, multiple slave processors are created. Each processor is given one rule for execution. It compares the activity with rule. If there is a match, then the alert is generated by that processor.

Since each processor is executing only one rule, the response time is improved very much. Once the anomaly in occupant activity is identified, the alert is identified by the search module. The alert message is also sent as SMS to occupant's care taker.

IV. EXPERIMENTAL RESULTS

The prototype is developed using *C#.net* and made to run on *Intel Pentium Dual Core machine*. For parallelization, the search module is implemented using *MPI.net* and to create virtual processors on a standalone machine *Microsoft Compute Cluster Pack* is installed. The Distributed *P* System is populated with 50 rules. Hence, 50 processors are created, and each processor is assigned a rule. The activity is compared with the rule assigned to each processor. The results obtained with single processor and with 50 processors are listed *Table 1*. Let T_s and T_m be the response time of the alert with a single processor and multiple processors respectively. Normally, Speedup (S) is computed as T_s/T_m . Hence from the *Table I*, it is proved experimentally, that due to parallelization, the Speedup is about 200(approx). This proves that decision making can be made faster thereby producing quick response in the smart spaces.

TABLE I. SPEED UP COMPUTATION

Abnormal Activity for which alert is generated	Response time with single processor (milliseconds) (T_s)	Response time with multiple processor (milliseconds) (T_m)	Speedup (S)
Watching TV – without spectacles	1.2066	0.0065	185.6
Sockets – wet hands	1.1366	0.0065	174.8
Walking – fast	0.9934	0.0051	194.7
Music System – very high volume	0.9971	0.0051	195.5

CONCLUSIONS

This paper focuses on the effective representation of the smart environment for efficient real time response. A domain specific application for assisted living in smart home has been proposed. Distributed *P* systems are generally used in representing the functioning of the system and thereby solving problems in a distributed manner. Algorithms are designed for representing smart spaces using Distributed *P* system. Hash table data structure is used for efficient representation of the data. This research work has aimed in satisfying the main need of smart environments by providing quick response for decision making and improves the response time by producing fast alerts using a parallelized approach.

REFERENCES

- [1] Vivek Menon, Bharat Jayaraman and Venu Govindaraju, "Three R's of Cyberphysical Spaces," IEEE Computer Society, vol. 44, no. 9, pp. 73-79, 2011.
- [2] Vivek Menon, Bharat Jayaraman and Venu Govindaraju, "Multimodal identification and tracking in smart environments," Personal and Ubiquitous Computing Journal-Springer, vol. 14, no. 8, pp. 685-694, 2010.
- [3] Vivek Menon, Bharat Jayaraman and Venu Govindaraju, "Biometrics driven smart environments: abstract framework and evaluation," Ubiquitous Intelligence and Computing-Springer, vol. 5061, pp. 75-89, 2008.
- [4] Vivek Menon, Bharat Jayaraman and Venu Govindaraju, "Integrating recognition and reasoning in smart environments," In Intelligent Environments, 4th International Conference, pp. 1-8, 2008.
- [5] Gheorghe Paun, Mario J. Perez-Jimenez, "Solving Problems in a Distributed way in Membrane Computing: dP Systems," International Journal of Computers, Communication and Control, vol. 2, pp. 238-250, 2010.
- [6] Susan Elias, Rajalakshmi V and Sivaranjani S, "Representation of Smart Environments using Distributed P Systems," In the Proceedings of Third International Conference on Advances in Communication, Network and Computing, pp. 159-164, 2012.